FACULTY WORKING PAPER 91-0173



# The IUR Methodology: Managing Uncertainty in Rule Induction



*Yong Ma*
*Department of Computer Science*
*University of Illinois*

*John S. Chandler*
*Department of Accountancy*
*University of Illinois*

Bureau of Economic and Business Research
College of Commerce and Business Administration
University of Illinois at Urbana-Champaign

# BEBR

FACULTY WORKING PAPER NO. 91-0173

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

October 1991

# The IUR Methodology: Managing Uncertainty in Rule Induction

Yong Ma
Department of Computer Science

John S. Chandler
Department of Accountancy

# The IUR Methodology: Managing Uncertainty in Rule Induction

Yong Ma
Department of Computer Science

John S. Chandler
Department of Accountancy

## Abstract

All inductive learning systems (ILSs) have to face the problem of handling uncertainty in data; in the data that is used to induce rules, the data that is classified by the rules, or both. Some ILSs cannot handle uncertainty at all, while others initially ignore uncertainty and then modify their original approaches to retroactively account for uncertainty. In either case, performance is effected and the resulting ILSs are generally, not theoretically consistent. To address the data uncertainty problem, this research assumes that uncertainty is inherent in all data and that a methodology cannot ignore its presence. The IUR (Induction of Uncertain Rules) methodology is developed which maintains data uncertainty until prediction computations must be made, resolving it through Dempster-Shafer combinations, thus producing a dynamically applicable inductive learning system. Besides yielding comparable classification performance, the IUR approach can produce probability bounds on its classifications and is linear in time complexity. This paper presents the details of the IUR methodology, provides a step by step example of the procedure, discusses the advantages and disadvantages of IUR, compares it to other well-known ILSs, and identifies future areas of research.

# The IUR Methodology: Managing Uncertainty in Rule Induction

## I. Introduction

All inductive learning systems (ILSs) have to face the problem of handling uncertainty in data; in the data that is used to induce rules, the data that is classified by the rules, or both. Some ILSs cannot handle uncertainty at all, while others initially ignore uncertainty and then modify their original approaches to retroactively account for uncertainty. In either case, performance is effected and the resulting ILSs are generally, not theoretically consistent. This research assumes that uncertainty is inherent in all data and then develops a methodology that maintains that data uncertainty until prediction computations must be made. The uncertainty is then resolved through Dempster-Shafer combinations, thus producing a dynamically applicable inductive learning system. Besides yielding comparable classification performance, the IUR (Induction of Uncertain Rules) approach can produce probability bounds on its classifications and is linear in time complexity. This paper presents the details of the IUR methodology, analyzes it in comparison to other well-known ILSs, and identifies future areas of research.

## II. Impetus

Any ILS that is to be applied to a real problem will face uncertainty in data. Uncertainty, for our purposes, includes missing values, erroneous values (or noise), and distributional dependencies. The data set upon which the classification rules are induced may contain uncertainty, which may produce faulty, or at least, ineffective, rules. The data set to which the induced rule is applied may also contain uncertainty, producing erroneous results, or, in some cases, no result.

Different ILSs handle uncertainty differently. In the case of missing values, some ILSs just stop (Quinlan, 1986a; Michalski et al, 1986; Clark & Niblett, 1989), while others use heuristics to determine a substitute value (Quinlan, 1986b, 1989; Mingers, 1989a, 1989b). In the case of noise, many ILSs initially assume noiseless data, and develop an appropriate heuristic or algorithm to handle such data (Michalski, 1986; Chan, 1989; Clark & Niblett, 1989). When noise is encountered, again, some ILSs may stop because they have reached an unknown state (early ID3s). Others have modified their initial approach by including additional modules just for noise (Quinlan, 1986b). The result can be viewed as a series of patches upon patches, with each

patch trying to address the inadequacies of the previous patch. The theoretically consistent initial system becomes an inconsistent kludge. As more problems and inadequacies are met, these approaches just continue to build even more complex and ad hoc ILSs.

The goal of this research is to develop a methodology that takes a general approach to the problem of uncertainty in data (as well as other problems of ILSs). The approach assumes that data that is certain can be viewed as a special case of the more generic case of uncertain data (i.e., when uncertainty equals 0), as opposed to viewing uncertain data as an anomaly of the default case of certain data. An analogy is the view that Bayesian combination is a special case of Dempster-Shafer combination, as opposed to the Dempster-Shafer approach being an extension of the Bayesian approach. From this top down view, all data is treated in the same, consistent manner and from the same theoretical background, as opposed to having certain data treated one way, uncertain data another, and then trying to combine them in a third way.

## III. The General Approach

The main difference between the IUR methodology and other ILSs is the timing of the resolution of uncertainty. Some ILSs ignore uncertainty in the beginning and then try to account for it later (Quinlan, 1986b), yielding the theoretically inconsistent hybrid systems. Others acknowledge the uncertainty but make probabilistic simplifications early in the approach, which make computations easier, but loses much of the significance (and risk and information) of the uncertainty (Rendell et al, 1987). The IUR methodology retains uncertainty in the data until the time a "rule" needs to be induced. It will then dynamically determine the impact of uncertainty through a series of Dempster-Shafer combinations, for that instantiation of the rule.

Uncertainty management is usually the domain of rule-based knowledge systems as opposed to ILSs. Many decades of analyses and experimentation have yielded several effective approaches such as MYCIN (Buchanan & Shortliffe, 1984), Dempster-Shafer (Dempster, 1967; Shafer, 1976), and so on. Before developing new techniques or heuristics, an investigation of the merits of proven techniques should be done first. The application of one of these proven techniques, Dempster-Shafer theory, to uncertainty management in an ILS, is the first such application.

# IV. The IUR Methodology

## A. Procedural Description

The IUR methodology is based on a decomposed, independent analysis of attributes against the training example set. A probability distribution analysis is made, attempting to preserve as much of the uncertainty as possible. The result is a array of probabilistic measures for all attributes. To classify a new example, IUR combines only those attributes that are affected by the example, using Dempster-Shafer conventions. The Dempster-Shafer analysis also produces probabilistic bounds for the predicted class. The IUR methodology can be summarized as follows.

Assume the following notation:

A set of attributes A.i, i=1,I
  where each A.i has a set of V.i values
    A.i = a(i,1), ..., a(i,v), ..., a(i,V.i)
      for v= 1,V.i,
    if A.i is a discrete attribute then V.i equals
      the number of discrete values
    if A.i is a continuous attribute then V.i equals
      the number of distinct values for A.i in
      K training examples (V.i <= K)
    and the generic value of A.i will be denoted
      as a.i instead of a(i,v), for simplicity.
A set of classes C.j, j=1,J.
A set of training examples T.k ,k=1,K,
  where T.k is a vector, [a(1,k), ..., a(I,k), c(k)]
    and a(i,k) is the value of attribute i for
      training example k and
    c(k) is the class value for training example k.
A set of unseen examples, E.u, u=1,U
  where E.u is a vector defined similar to T.k,
    E.u = [a(1,u), ..., a(I,u), c(u)]
Note: Any attribute value a(i,k) or a(i,u) can be missing
  and have a null value.

Phase 1. Generate probability assignment distribution of all attributes, A.i.

**Step 1.** Analyze the distribution of class values across attribute values. For continuous attributes, sort in increasing value to produce a table similar to the following:

| | Number of examples in each class for each a.i | | | | | | |
|---|---|---|---|---|---|---|---|
| a.i | C.1 | C.2 | C.3 | ... | C.j | ... | C.J |
| 3 | 2 | 0 | 0 | ... | 0 | ... | 0 |
| 4 | 2 | 0 | 0 | ... | 0 | ... | 0 |
| 8 | 2 | 0 | 0 | ... | 0 | ... | 0 |
| 11 | 2 | 1 | 0 | ... | 0 | ... | 0 |
| 37 | 0 | 8 | 8 | ... | 0 | ... | 0 |
| 106 | 0 | 1 | 0 | ... | 0 | ... | 0 |
| 129 | 0 | 4 | 0 | ... | 0 | ... | 0 |

**Step 2.** Determine brackets (ranges) of a.i values that yield the same distribution of class values.

A bracket r for attribute i, b[i,r], is defined as having particular a.i values that will be called the bracket's raw lower bound, L[i,r]', and upper bound, U[i,r]'. The simplest method for determining a bracket from training examples is to define a new bracket (i.e., new lower and upper bounds) when there is a change in the class value or there is more than one class value for a.i. The particular values of L[i,r]' and U[i,r]' are identified as the longest interval of a.i such that there is no change in class values when a.i goes from L[i,r]' to U[i,r]'. Each attribute will have its own set of B.i brackets.

Using the example in Step 1, the distribution of examples across class values remains the same for a.i values of 3, 4, and 8. When a.i = 11, however, the distribution of examples across class values changes. Thus, the first bracket for attribute i, b[i,1], starts with a.i = 3 and ends between a.i = 8 and a.i = 11. The complete set of brackets for the example would be:

| | bracket bounds | | Number of examples in each class for a.i | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| bracket | L[i,r]' range | U[i,r]' range | C.1 | C.2 | C.3 | ... | C.j | ... | C.J |
| b[i,1] | 3 | 8-11 | 6 | 0 | 0 | ... | 0 | ... | 0 |
| b[i,2] | 8-11 | 11-37 | 2 | 1 | 0 | ... | 0 | ... | 0 |
| b[i,3] | 11-37 | 37-106 | 0 | 8 | 8 | ... | 0 | ... | 0 |
| b[i,4] | 87-106 | 129 | 0 | 5 | 0 | ... | 0 | ... | 0 |

There are many other ways to define the criteria for a bracket. As an example, one can break on a minimum (maximum) number of examples for a range of values of a.i, e.g., each bracket must have at least (at most) 10 examples. Or, one can break on a proportional basis, e.g., a bracket is formed when the proportion of examples in one class exceeds 90% (a dominance criteria). There are also many methods to define the upper and lower bounds for a bracket in terms of a.i for the purpose of

classifying unseen examples, i.e., L[i,r] and U[i,r]. Section IV.C details five methods for determining these bounds.

**Step 3.** Determine the m value, probability assignment in Dempster-Shafer terms, for each bracket within an attribute. There are many methods for calculating m, ranging from standard probability theory to Dempster-Shafer-based computations. Section IV.D. presents four different methods. The result of this transformation of frequency counts to m values is a 2-dimensional array, [B.i x (J+1)], the probability assignment array (PAA), for each attribute, i.e., PAA(i).

m(i,r,j) = m value for class j, within bracket r, for attribute i

THETA = defined in Dempster-Shafer terms as the unassigned portion of the belief function

For A.i, PAA(i):

| | C.1 | C.2 | ... | C.J | THETA |
|---|---|---|---|---|---|
| b[i,1] | m(i,1,1) | m(i,1,2) | ... | m(i,1,J) | 1-SUM |
| b[i,2] | m(i,2,1) | m(i,2,2) | ... | m(i,2,J) | 1-SUM |
| | | | ... | | |
| b[i,B.i] | m(i,B.i,1) | m(i,B.i,2) | ... | m(i,B.i,J) | 1-SUM |

The resulting PAA(i)s retain much of the distributional properties of the training examples. A decision "rule" has not been induced yet, and in terms of the overall IUR approach, a static rule is never actually induced. Instead, these arrays, which can be called rule sets, are used to produce a classification prediction for a new, unseen example; essentially, inducing a "rule" dynamically for each example. As will be discussed later, this process is linear in the number of attributes (I) and number of classes (J), which is as fast as other approaches that induce a static rule.

Phase 2. Classify new, unseen examples by a series of Dempster-Shafer combinations on the PAA

For each unseen examples, E.u
where E.u is a vector,
E.u = [a(1,u), ..., a(i,u), ..., a(I,u), c(u)]
and a(i,u) is the value of attribute i for the unseen example u and c(u) is the class value for unseen example u.

Step 1. For each a(i,u), find the bracket in PAA(i) that contains the value of a(i,u), call it B(i.u). Each B(i.u) is vector of (J+1) m values, one for each of the J class values and one for THETA. If an attribute is missing from E.u, then no vector B(i.u) is found and used for that attribute.

Step 2. Combine the m values across all the B(i.u) vectors using Dempster-Shafer combinations, to yield a final prediction vector, P(i.u), which contains the (J+1) combined m values for each class value and THETA.

Step 3. Find the maximum m value within P(i.u), and predict the associated class value (or THETA).

The above three step process for prediction is done for each unseen example. Depending on the individual values of each a(i,u) in E.u, a different set of B(i.u) vectors may be used for each unseen example. From the definition of m, one can also derive Bel and Pl. Thus, P(i.u) not only provides a means for predicting a class value for an unseen example, but it also provides the data to determine, (m, Bel, Pl) for each class and THETA for that unseen example.

B. Determination of Brackets

To complete the definition of a bracket for classification purposes, L[i,u] and U[i,r] must be determined. Because there may be gaps in terms of a.i, between U[i,r]' and L[i,r+1]', a consistent method must be adopted to determine L[i,r] and U[i,r]. Five methods are presented below. The first method retains the gaps, while the other four methods make assumptions about how to cover the gaps. Two conventions for all methods are (1) L[i,1] = min[a.i] and (2) U[i,B.i] = max[a.i]. The example data from Section IV.A. will be used to demonstrate each of the five methods. From that example, the bracket boundaries to be determined are between 8-11, 11-37, and 37-106. The following data are also necessary to compute boundary values. Figure 1 shows how each of the five methods for bracketing A.i.

| Bracket | L[i,r]' | U[i,r]' |
|---|---|---|
| b[i,1] | 3 | 8 |
| b[i,2] | 11 | 11 |
| b[i,3] | 37 | 37 |
| b[i,4] | 106 | 129 |

Method 1: L[i,r] = L[i,r]'
U[i,r] = U[i,r]'

b[i,1]: 3-8
b[i,2]: 11-11
b[i,3]: 37-37
b[i,4]: 106-129

Method 2: L[i,r] = U[i,r-1]' + 1
U[i,r] = U[i,r]'

b[i,1]: 3-8
b[i,2]: 9-11
b[i,3]: 12-37
b[i,4]: 38-129

Method 3: L[i,r] = L[i,r]'
U[i,r] = L[i,r+1]' - 1

b[i,1]: 3-10
b[i,2]: 11-36
b[i,3]: 37-105
b[i,4]: 106-129

Method 4: L[i,r] = U[i,r-1] + 1
U[i,r] = FL{(U[i,r]' + L[i,r+1]')/2}

b[i,1]: 3-FL{(8+11)/2} 3-9
b[i,2]: (9+1)-FL{(11+37)/2} 10-24
b[i,3]: (24+1)-FL{(37+106)/2} 25-71
b[i,4]: (71+1)-129 71-129

Method 5: L[i,r] = U[i,r-1]' + 1
U[i,r] = U[i,r]' +
$$FL\left|(L[i,r+1]'-U[i,r]')*\frac{W\{U[i,r]'\}}{W\{U[i,r]'\}+W\{L[i,r+1]'\}}\right|$$

where W{U[i,r]'} is the number of training examples for a.i that equals U[i,r]
and W{L[i,r]'} is the number of training examples for a.i that equals L[i,r]

b[i,1]: 3-[8+FL{(11-8)*(2/(2+3))}]
3-[8+FL{1.2}]

```
b[i,2]: (9+1)-[11+FL{(37-11)*(3/(3+16))}]
        10-[11+FL{4.1}]                                  10-15

b[i,3]: (15+1)-[37+FL{106-37)*(16/(16+1))}]
        16-[37+FL{64.8}]                                 16-111

b[i,4]: (111+1)-129                                     112-129
```

C. Calculation of m

Ma and Wilkins (1990) proposed four methods for calculating m. These are summarized below. Analysis of the four methods is found in Ma and Wilkins (1990). Assume that there exists a set of training examples of solved cases, and that the goal is to construct a rule set that maps evidence to hypotheses about a given class C.j. Also assume that for a given attribute a.i, there are B.i brackets b[i,r]. Let

p(i,r) = number of positive instances in b[i,r] for class C.j
n(i,r) = number of negative instances in b[i,r] for class C.j
P(i) = total number of positive instance for class C.j, across all brackets, SUM.r(p(i,r))
N(i) = total number of negative instance for class C.j, across all brackets, SUM.r(n(i,r))
m = basic probability assignment for b[i,r] for class C.j

where a "positive instance" is an instance whose classification is C.j, and a "negative instance" is, thus, an instance whose classification is not C.j. In the following discussion, the set {C.j, NOT C.j} denotes the hypothesis space in which rules are induced from the training set, and THETA is the unassigned portion of the Dempster-Shafer belief function. The following table identifies the values required for each of the four methods.

| Bracket | C.j | NOT C.j (all other classes) | {C.j, NOT C.j} |
|---|---|---|---|
| b[i,1] | p(i,1) | n(i,1) | p(i,1)+n(i,1) |
| b[i,2] | p(i,2) | n(i,2) | p(i,2)+n(i,2) |
| . . . | | | |
| b[i,r] | p(i,r) | n(i,r) | p(i,r)+n(i,r) |
| . . . | | | |
| b[i,B.i] | p(i,B.i) | n(i,B.i) | p(i,B.i)+n(i,B.i) |
| | P(i) | N(i) | P(i)+N(i) |

Method 1

$$m(C.j) = \frac{p(i,r)}{p(i,r)+n(i,r)}$$

$$m(\text{NOT } C.j) = \frac{n(i,r)}{p(i,r)+n(i,r)}$$

$$m(\text{THETA}) = 0$$

This method follows standard probability theory. Its most significant drawback is that if a bracket contains only positive or only negative instances then either m(C.j) = 1 or m(NOT C.j) = 1, a very strong statement in either standard probability theory or Dempster-Shafer theory.

Method 2

$$m(C.j) = \frac{p(i,r)}{P(i)+N(i)}$$

$$m(\text{NOT } C.j) = \frac{n(i,r)}{P(i)+N(i)}$$

$$m(\text{THETA}) = 1 - m(C.j) - m(\text{NOT } C.j)$$

This is a slight improvement over method 1 but it still has a problem when p(i,r), P(i), n(i,r), and N(i) are all approximately equal, i.e., when one bracket contains almost all of the instances, and they are evenly split between positive and negative instances. This situation yields m(C.j) and m(NOT C.j) roughly equal to .5 and m(THETA) = 0. What one would rather have is m(C.j) and m(NOT C.j) close to 0 and m(THETA) close to 1, indicating that a bracket with an even split of instances has poor discriminatory power.

Method 3

$$m(C.j) = \begin{cases} \frac{p(i,r)-n(i,r)}{P(i)} & \text{if } p(i,r) \geq n(i,r) \\ 0 & \text{otherwise} \end{cases}$$

$$m(\text{NOT } C.j) = \begin{cases} \frac{n(i,r)-p(i,r)}{N(i)} & \text{if } n(i,r) \geq p(i,r) \\ 0 & \text{otherwise} \end{cases}$$

$$m(\text{THETA}) = 1 - m(C.j) - m(\text{NOT } C.j)$$

The difference $|p(i,r)-n(i,r)|$ introduced in this method, makes a rule that covers an almost equal number of positive and negative

instances distinct from a rule that covers an unequal number of positive and negative instances.

Method 4

$$m(C.j) = \begin{cases} \dfrac{p(i,r)-n(i,r)}{P(i)} & \text{if } p(i,r)>=n(i,r) \\ \dfrac{p(i,r)}{P(i)} \dfrac{1}{N(i)[n(i,r)-p(i,r)]} & \text{otherwise} \end{cases}$$

$$m(\text{NOT } C.j) = \begin{cases} \dfrac{n(i,r)-p(i,r)}{N(i)} & \text{if } n(i,r)>=p(i,r) \\ \dfrac{n(i,r)}{N(i)} \dfrac{1}{P(i)[p(i,r)-n(i,r)]} & \text{otherwise} \end{cases}$$

$$m(\text{THETA}) = 1 - m(C.j) - m(\text{NOT } C.j)$$

Method 3 has a strong bias to set m(NOT C.j) to 0 if p(i,r)>n(i,r). This is undesirable if n(i,r) is large. Method 4 corrects this situation and gives some credit to n(i,r) if n(i,r) > 0.

D. Complexity Considerations

Time complexity is always a concern with any ILS. Barnett (1981, 1991) presented an efficient method (i.e., linear complexity) to compute Bel and Pl for Dempster-Shafer theory when bodies of evedence focus on basic hypotheses and their complements. Here, we use an even simpler and more efficient computational procedure to combine evidence in Dempster-Shafer theory. Therefore, IUR is linear in all aspects of its processing. Determination of the brackets in Phase 1 is linear in terms of the number of training examples (K) and number of attributes (I), i.e., O(K*I). Creation of PAA is linear in terms of the number of brackets in PAA and the number of classes (C). Each attribute i has its own number of brackets, B.i, and the total number of brackets is SUM(B.i). Thus, creation of PAA is of the order O(SUM(B.i)*C)). Phase 2 computations are also linear in terms of the number of attributes (I) and number of classes (C), thus, O(I*C). The Dempster combination process is linear in terms of the number of classes, C, thus, O(C).

## V. An Example

To demonstrate the IUR methodology we use the classic IRIS classification problem of Fisher (1936). As the data set is well-known and well used, we do not include the data in this paper. There are four continuous attributes, sepal length (seplen), sepal width (sepwid), petal length (petlen), and petal width (petwid); and three classes setosa (set), versicolour (ver), and virginica (ver). We will go through each phase and step of the process.

### A. Phase 1

#### 1. Step 1 and Step 2

The goal of these two steps is to transform the raw training examples into a set of bracket definitions, for each attribute. In this example we defined the bounds of a bracket using Method 3 from section IV.B. The resulting brackets for each attribute are shown in Table I.

#### 2. Step 3

The goal of this step is to translate the distributional data of the bracket arrays into m values for each bracket. For this example we used the Method 3 from section IV.C. formulation for m. The resulting PAA is shown in Table II.

### B. Phase 2

To demonstrate the process of IUR we have chosen four examples from the IRIS dataset.

Example 1: An obvious correct prediction (correct class = set)

**Step 1:**

| Attribute | raw value | bracket | m values (set, ver, vir, theta) |
|---|---|---|---|
| sepal length | 57 | 57-57 | ( .000, .040, .000, .960) |
| sepal width | 38 | 38-38 | ( .040, .000, .000, .960) |
| petal length | 17 | 10-29 | (1.000, .000, .000, .000) |
| petal width | 03 | 01-09 | (1.000, .000, .000, .000) |

**Step 2:**

( .000, .040, .000, .960) D+S ( .040, .000, .000, .960)
= ( .038, .038, .000, .924)

( .038, .038, .000, .924) D+S (1.000, .000, .000, .000)
= (1.000, .000, .000, .000)

(1.000, .000, .000, .000) D+S (1.000, .000, .000, .000)
= (1.000, .000, .000, .000)

Step 3:
Thus, because class 1 is 1.000, the predicted class is the first class, set; which is correct. Note THETA is 0.

Example 2: A weak correct prediction (correct class = vir)

| Step 1: Attribute | raw value | bracket | m values (set, ver, vir, theta) |
|---|---|---|---|
| sepal length | 63 | 63-63 | ( .000, .000, .060, .940) |
| sepal width | 28 | 28-28 | ( .000, .000, .040, .960) |
| petal length | 51 | 51-51 | ( .000, .000, .120, .880) |
| petal width | 15 | 15-15 | ( .000, .160, .000, .840) |

Step 2:
( .000, .000, .060, .940) D+S ( .000, .000, .040, .960)
= ( .000, .000, .098, .902)

( .000, .000, .098, .902) D+S ( .000, .000, .120, .880)
= ( .000, .000, .206, .794)

( .000, .000, .206, .794) D+S ( .000, .160, .000, .880)
= ( .000, .131, .179, .690)

Step 3:
Thus, because .179 is the maximum m value for the class values, the predicted class is the third class, vir; which is correct. Note, however, the high value of THETA.

Example 3: An incorrect prediction (correct class = ver)

| Step 1: Attribute | raw value | bracket | m values (set, ver, vir, theta) |
|---|---|---|---|
| sepal length | 60 | 60-60 | ( .000, .040, .000, .960) |
| sepal width | 27 | 27-27 | ( .000, .020, .000, .980) |
| petal length | 51 | 51-51 | ( .000, .000, .120, .880) |
| petal width | 16 | 16-16 | ( .000, .040, .000, .960) |

Step 2:
( .000, .040, .000, .960) D+S ( .000, .020, .000, .980)
= ( .000, .059, .000, .941)

( .000, .059, .000, .941) D+S ( .000, .000, .120, .880)
= ( .000, .052, .114, .834)

( .000, .052, .114, .834) D+S ( .000, .040, .000, .960)
= ( .000, .086, .110, .804)

**Step 3:**

Thus, because .110 is the maximum m value for the class values, the predicted class is the third class, vir; which is incorrect. Note the high value for THETA.

Example 4: Missing attribute (note, class = vir)

4a: Calculations with all attributes present

**Step 1:**

| Attribute | raw value | bracket | m values (set, ver, vir, theta) |
|---|---|---|---|
| sepal length | 58 | 58-58 | ( .000, .000, .000,1.000) |
| sepal width | 27 | 27-27 | ( .000, .020, .000, .980) |
| petal length | 51 | 51-51 | ( .000, .000, .120, .880) |
| petal width | 19 | 19-25 | ( .000, .000, .680, .320) |

**Step 2:**

( .000, .000, .000,1.000) D+S ( .000, .020, .000, .980)
= ( .000, .020, .000, .980)

( .000, .020, .000, .980) D+S ( .000, .000, .120, .880)
= ( .000, .018, .118, .864)

( .000, .018, .118, .864) D+S ( .000, .000, .680, .320)
= ( .000, .006, .714, .280)

**Step 3:**

Thus, because .714 is the maximum m value of the class values, the predicted class is the third class, vir; which is correct. Note low value for THETA.

4b: Calculation with attribute four (petwid) missing

**Step 1:**

| Attribute | raw value | bracket | m values (set, ver, vir, theta) |
|---|---|---|---|
| sepal length | 58 | 58-58 | ( .000, .000, .000,1.000) |
| sepal width | 27 | 27-27 | ( .000, .020, .000, .980) |
| petal length | 51 | 51-51 | ( .000, .000, .120, .880) |
| petal width | - | - | - |

Step 2:

( .000, .000, .000,1.000) D+S ( .000, .020, .000, .980)
= ( .000, .020, .000, .980)

( .000, .020, .000, .980) D+S ( .000, .000, .120, .880)
= ( .000, .018, .118, .864)

Step 3:

Thus, because .118 is the maximum m value of the class values, the predicted class is the third class, vir; which is correct. Note high value for THETA

## 4c: Calculation with mean value substituted for missing attribute four (petwid)

Step 1:

| Attribute | raw value | bracket | m values (set, ver, vir, theta) |
|---|---|---|---|
| sepal length | 58 | 58-58 | ( .000, .000, .000,1.000) |
| sepal width | 27 | 27-27 | ( .000, .020, .000, .980) |
| petal length | 51 | 51-51 | ( .000, .000, .120, .880) |
| mean petwid | 12 | 10-13 | ( .000, .560, .000, .440) |

Step 2:

( .000, .000, .000,1.000) D+S ( .000, .020, .000, .980)
= ( .000, .020, .000, .980)

( .000, .020, .000, .980) D+S ( .000, .000, .120, .880)
= ( .000, .018, .118, .864)

( .000, .018, .118, .864) D+S ( .000, .560, .000, .440)
= ( .000, .537, .056, .407)

Step 3:

Thus, because .537 is the maximum m value of the class values, the predicted class is the second class, ver; which is incorrect. Thus, by substituting the mean value for the missing attribute an incorrect prediction was made, whereas ignoring the attribute (example 4b) managed to produce a correct prediction.

## C. Post classification analysis of examples

One of the advantages of using the IUR methodology is the wealth of information it provides about the classification of an example. By definition, determining m also determines Bel and Pl, the lower and upper bounds on m. The table below provides

Bel and Pl for each of the examples in section V.B. (C= correct class and * = IUR prediction).

| Example | Class | m | Bel | Pl |
|---|---|---|---|---|
| 1:Correct | set | 1.000*C | 1.000 | 1.000 |
| | ver | 0.000 | 0.000 | 0.000 |
| | vir | 0.000 | 0.000 | 0.000 |
| | THETA | 0.000 | 1.000 | 1.000 |
| 2:Correct | set | 0.000 | 0.000 | 0.690 |
| | ver | 0.131 | 0.131 | 0.821 |
| | vir | 0.179*C | 0.179 | 0.869 |
| | THETA | 0.690 | 1.000 | 1.000 |
| 3:Incorrect | set | 0.000 | 0.000 | 0.804 |
| | ver | 0.086 C | 0.086 | 0.890 |
| | vir | 0.110* | 0.110 | 0.914 |
| | THETA | 0.804 | 1.000 | 1.000 |
| 4a:Correct | set | 0.000 | 0.000 | 0.280 |
| | ver | 0.006 | 0.006 | 0.284 |
| | vir | 0.714*C | 0.714 | 0.994 |
| | THETA | 0.280 | 1.000 | 1.000 |
| 4b:Correct (with missing attribute) | set | 0.000 | 0.000 | 0.864 |
| | ver | 0.018 | 0.018 | 0.882 |
| | vir | 0.118*C | 0.118 | 0.982 |
| | THETA | 0.864 | 1.000 | 1.000 |
| 4c:Incorrect (with average value) | set | 0.000 | 0.000 | 0.407 |
| | ver | 0.537* | 0.537 | 0.944 |
| | vir | 0.056 C | 0.056 | 0.463 |
| | THETA | 0.407 | 1.000 | 1.000 |

In Section V.B. class predictions were based on the highest value of m. Because IUR can provide the above data, other selection criteria can be used. If a decision maker is risk averse, then class prediction would be based on max(Bel), the highest lower bound, indicating the highest chance of non-failure. If a decision maker is risk seeking, then class prediction would be based on min(Pl), the lowest upper bound, indicating the highest chance of failure. The distance between Bel and Pl can also be used as a criteria, with a smaller distance implying a smaller chance for error. Ideally, the selection should choose the option with the highest Bel and highest Pl, but that combination does not always occur. To determine an appropriate choice in this case, a heuristic method has been proposed (Ma et al, 1991).

# VI. Analysis of the IUR Methodology

## A. Advantages

The initial impetus for this research was the inability (or ineffectiveness) of ILSs to handle uncertainty. The IUR methodology addresses uncertainty by retaining it in the m values of the PAA until an actual classification determination is needed. Then the uncertainty is resolved using repeated Dempster-Shafer combinations. No ad hoc or heuristic procedures need to be tacked on to the basic approach to handle the uncertainty.

Missing values are a particular problem for ILSs, either causing them to stop or substitute artificial values. IUR ignores missing values (Phase 2, Step 1). This is theoretically consistent in that if there is no evidence available for an attribute, i.e., a missing attribute, then no evidence should be used (e.g., by substitution). Missing values do not stop the IUR process, but actually speed the process, which is the opposite effect when other ILSs must find reasonable substitutes.

As will be shown in the graphical analysis of IUR in Section VII.A., many ILSs lose information on the distribution of attribute values when they are forced to make a static rule. As a result outliers can have a greater effect than deserved in rule induction. "Fixes" would include washing the data through a statistical outlier program before inducing a rule, again mixing approaches. IUR retains much of this distributional information through the m values.

The computational linearity of IUR, discussed in section IV.D., provides another benefit. Efficient incremental learning is a very important aspect of ILSs (Utgoff, 1988, 1989; Schlimmer & Fisher, 1986). In IUR, incremental learning is very efficient. To add new examples to the training example set Phase 1 is recalculated, possibly yielding new brackets, but always yielding a new PAA. As the "rule" induction phase is only done when required, incorporation of new examples is simple.

The use of Dempster-Shafer analysis not only provides a way for handling uncertainty to classify examples, but also yields additional information on the confidence of that classification. m values can be transformed into Bel and Pl values, thus, giving lower and upper bounds on the predicted classifications. This can be used to present an order list of classifications, in additional to a single class. As P(i.u) contains m values for all classes and THETA, IUR can output an array of results which can allow the decision maker to choose a different class than the one with the maximum m value. Not all ILSs allows such flexibility.

Irrespective of the definition of selection criteria, whether it be one of the above or a combination of several, the point is that IUR allows the decision maker the opportunity to define their own criteria. With different set of data, a decision maker may have different sets of expectations and confidence. For very noisy domains, a decision maker may be conservative and risk averse. For very clean, deterministic domains, however, a decision maker may wish to put tight controls over predictions and want to use a distance measure. Over time, a decision maker may become more confident in the application of IUR, and wish to relax the bounds on prediction. Data may also stabilize (or destabilize) over time. IUR does not prevent any of these cases. The beauty of IUR is that, in contrast to other ILSs, it does not make a prediction, it provides the decision maker with a wealth of information about the prediction environment and allows the decision maker to choose.

### B. Disadvantages

The major practical disadvantage of IUR is that no static decision rule is produced. There is nothing for a decision maker to look at and use. The PAA, which is essentially a rule set, can be very large, and would be incomprehensible to most users. The only counter to this criticism is that the advantages listed above are great enough to overcome this deficiency.

Another practical disadvantage is the lack of insight into the discriminatory importance of individual attributes. In discriminant analysis one can examine the coefficients to determine relative importance. In a TDDT, attributes not in the resulting tree can be eliminated from consideration, and analysis of the tree, weighted by number of cases per branch, can give the decision maker an idea of relative importance. Currently, one does not get an overall evaluation of an attribute's contribution to discrimination. This is an area of future investigation.

A more theoretical criticism is that the independent attribute analysis ignores the interdependencies among attributes. Such dependencies are most obvious in the graphical output of TDDTs. But this criticism is only superficial in that the dependencies still remain and are handled by the Dempster-Shafer combinations in step 2 of Phase 2.

## VII. Comparison to Other ILSs

The proof of any new approach to rule induction is in a comparison to existing methodologies. This section provides such a comparison, on both a graphical and empirical level. Neither comparison, however, is meant to be exhaustive, only illustrative. Future studies will explore the comparisons in

depth.

A. Graphical

To put IUR's approach in perspective we present a graphical interpretation of the IUR methodology, comparing it to those of a statistical approach (discriminant analysis) and a top-down decision tree (TDDT) approach (such as ID3). Assume the sample data distribution in Figure 2, in which there are two attributes, A.1 and A.2 (with values of a.1 and a.2), and two classes, x and o. Discriminant analysis attempts to define a line that separates the two clusters, minimizing erroneous classifications (Figure 3). A TDDT approach builds rectangular regions around clusters examples with similar class values (Figure 4). IUR defines brackets for each attribute, A.1 (Figure 5) and A.2 (Figure 6). Notice that a new bracket is formed whenever the distribution of class values changes. For example, in bracket b[1,1] distribution is all "x"s and no "o"s, whereas in b[1,2] the distribution is 3 "x"s and 1 "o."

One of the advantages of the IUR methodology is its ability to reflect the distributional characteristics of attribute values. Assume a similar, yet different, example data distribution wiht the addition of two X's and two O's (Figure 7). The TDDT approach yields the same set of rectangular regions (Figure 8), and, thus, will produce the same predicted values as the TDDT corresponding to Figure 4. IUR, however, yields the same set of brackets for A.1 but with different m values (Figure 9), and a different set of brackets and corresponding m values for A.2, (Figure 10). These brackets will eventually produce different predictions than those from Figures 5 and 6. This demonstrates that a TDDT approach treats every example as equal, within a rectangle, whereas IUR treats each example on its own merits.

Maintaining distributional characteristics also plays a significant role in incremental learning. If one assumes that the X's and O's in Figure 7 are incremental training examples, then IUR "learns" by changing its bracket definitions and corresponding m values. A TDDT, however, may not "learn" if the new training examples are covered by the existing rule. Thus, a TDDT approach may be slow to learn.

B. Empirical

To get a feel for the practical prowess of IUR, several classic data sets were analyzed. In addition to the Iris data set of Fisher (1936), used in the example in section V.B., three cancer datasets from the Institute of Oncology at Ljubljana, Yugoslavia, accumulated by Zwitter and Soklic were used: lymphography, breast cancer, and primary tumor. All four methods of determining m were employed for the IUR analysis. Results for

four other approaches (Assistant, Bayes, CN2, and AQ15) were compiled from the literature. The table below presents their respective error rates. (Note: The rates for IUR are classification rates.)

Error Rates

| | <-----IUR----> | | | | [1] | [1] | [2] | [3] | [3] |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | M1 | M2 | M3 | M4 | ASST | Bayes | CN2 | AQ15 | Experts |
| Iris | 3 | 3 | 3 | 3 | x | 7[4] | x | x | x |
| Lymphography | 19 | 45 | 36 | 36 | 24 | 17 | 18 | 19 | 15 |
| Breast Cancer | 29 | 30 | 30 | 30 | 22 | - | 31 | 31 | 36 |
| Primary Tumor | 75 | 75 | 70 | 66 | 56 | 52 | 55 | 65 | 58 |

[1] Cestnik et al (1986)
[2] Clark and Niblett (1987)
[3] Michalski et al (1986)
[4] Weiss and Kapouleas (1989)

As one can see, IUR has comparable or even better performance than other approaches, except for the primary tumor dataset. Further testing needs to be done, but, if accuracy and time complexity is comparable, the added benefits of IUR identified in section VI make IUR an alternative worth exploring.

## VIII. Future Research

This paper has presented a description of a new approach to managing uncertainty in inductive learning systems. Because of its introductory nature, this paper cannot go into all of the details and implications of the IUR approach. This section describes three areas of future research that will provide the depth and breadth required of a new methodology: a deeper analysis of current formulation of the methodology, extensions of the methodology in terms of theory and applicability, and rigorous comparative empirical studies.

### A. Deeper analysis of current methodology

One of the innovative aspects of the IUR methodology is the concept of bracketing the range of attribute values. Five different methods for determining brackets were presented in section IV.B., but the sensitivity of IUR performance to the choice of bracketing method must be investigated. The five presented methods are all dynamic, that is, the number of brackets per attribute is a function of the class value distributions within attribute value distributions. Alternative approaches include having a fixed number of brackets, irrespective of the distribution of values; basing brackets on

the attribute value distribution only (e.g., one bracket every standard deviation); or basing brackets on probabilistic measures such as m. All of these approaches need to be investigated theoretically and empirically.

Another important aspect of the IUR methodology is the determination of m. Four methods were presented in section IV.C. with variable results in section VI.B. Theoretical analyses has already been performed (Ma and Wilkins, 1990). A more rigorous and broad empirical evaluation, however, must be performed.

The current implementation of IUR relies heavily on the distribution of attribute values. Although (m, Bel, Pl) can characterize noise in the values, the sensitivity of IUR to typical distributional characteristics will be informative. For continuous-valued attributes, different distributions (e.g., normal, exponential, uniform, bimodal, and skewed) will be simulated to determine the response of IUR. Discrete-valued attributes, and combinations of discrete and continuous attributes, will also be investigated.

Dempster-Shafer theory is the generalization of Bayesian revision theory. Thus, IUR should have a natural link to Bayesian classifiers. A more formal theoretical comparison of the process and outcomes of IUR and Bayesian classifiers will be done.

One of the unique aspects of the IUR approach is its philosophy of carrying a significant portion of the distributional knowledge of the data (both certain and uncertain) through the process until computations are necessary. One implication is that other ILS approaches that do not retain this knowledge have, thus, lost information, which could have been used later in the induction process. A formal information theoretic analysis will be made with respect to information lost (1) by heuristics of other ILSs, (2) as a result of missing values, and (3) as result of the dominance of attributes. Such an analysis will help define the benefits of IUR.

And finally, the original impetus for developing IUR was to address the problems of noise in inductive learning. Thus, a natural follow-up study will be to replicate the noise studies of Quinlan (1986b) and the missing value studies of Mingers (1989a, 1989b). We expect to find less sensitivity and greater capability with IUR.

B. Extensions of methodology

Although IUR was developed to produce classification predictions, it may also provide an efficient environment for incremental learning as suggested in section VI.A. The ability of an ILS to learn and adapt to a changing data environment is a

crucial factor in determining the practicality of an approach. Much research (e.g., Utgoff, 1990) has gone into incremental learning, and an in-depth analysis of the applicability of IUR to this process is needed.

One of the main criticisms that can be raised against the IUR approach is that a comprehensible, static decision rule is not produced. In contrast, however, IUR does produce a wealth of information in terms of (m, Bel, Pl). Serious theoretical analysis needs to be done to investigate the development of confidence measures from (m, Bel, Pl). Different decision making criteria were mentioned in section V.C. The decision making heuristics developed in Ma et al (1991) needs to be expanded to a broader decision making framework. And, finally, the feasibility of developing a "comprehensibility" metric, based on (m, Bel, Pl) needs to be investigated, to parallel the comprehensibility of a TDDT.

C. Rigorous empirical comparison studies

The empirical analysis in section VI.B. was only for illustrative purposes. A more rigorous approach to empirical comparisons will be made, using the most current testing techniques such as the n-fold cross validation methods. Additional datasets will be used, not only the classic datasets, but new ones from business domains. Additional, or updated, versions of ILSs will complete the analysis. The goal will be to give a comprehensive evaluation of the advantages and disadvantages of the IUR methodology.

Cited References

Barnett, J.A. (1981) Computational Methods for a Mathematical Theory of Evidence, Proceedings of IJCAI-81, 868-875, Vancouver, BC; AAAI.

Barnett, J.A. (1991) Calculating Dempster-Shafer Plausibility, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13, 6, 599-602.

Buchanan, B.G., & Shortliffe, E.H. (1986) Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Reading, PA: Addison Wesley.

Cestnik, G., Konenenko, I., & Bratko, I. (1987) Assistant-86: A Knowledge-elicitation tool for Sophisticated Users. In I.Bratko and N. Lavrac (Eds.) Progress in Machine Learning, 31-45: Sigma Press.

Chan, P.K. (1989) Inductive Learning with BCT, Proceedings of the Sixth International Workshop on Machine Learning, 104-108, Ithaca, NY: Morgan Kaufmann.

Clark, P., & Niblett, T. (1987) Induction in Noisy Domains. Progress in Machine Learning (from the Proceedings of the Second European Working Session on Learning), 11-30, Bled, Yugoslavia: Sigma Press.

Clark, P., & Niblett, T. (1989) The CN2 Induction Algorithm, Machine Learning, 3, pp261-283.

Dempster, A.P. (1967) Upper and lower probabilities induced by a multivalued mapping, Annals of Mathematical Statistics, 38, 325-339.

Fisher, R. (1936) The Use of Multiple Measurements in Taxonomic Problems, Annals of Eugenics, 7, 179-188.

Ma, Y., Chandler, J.S., and Wilkins, D.C. (1991) On the Decision Making Problem in Dempster-Shafer Theory, Faculty Working paper #91-0172, College of Commerce and Business Administration, University of Illinois at Urbana, October 1991.

Ma, Y., & Wilkins, D.C (1990) Computation of Rule Probability Assignments for Dempster-Shafer Theory and the Sociopathicity of the Theory, Proceedings of the Third International Symposium on Artificial Intelligence, Mexico, 214-219.

Michalski, R.S., Mozetic, I., Hong, J., and Lavrac, N. (1986). The Multi-purpose incremental learning system AQ15 and its testing application to three medical domains, Proceedings of the Fifth National Conference on Artificial Intelligence, 1041-1045, Philadelphia, PA: Morgan Kaufmann.

Mingers, J. (1989a) An Empirical Comparison of selection measures for decision-tree induction, Machine Learning, 3, 319-342.

Mingers, J. (1989b) An Empirical Comparison of pruning methods for decision-tree induction, Machine Learning, 4, 227-243.

Quinlan, J.R. (1986a), Induction of decision trees, Machine Learning, 1, 81-106.

Quinlan, J.R. (1986b), The effect of noise on concept learning, In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell, (Eds.), Machine Learning: An artificial intelligence approach (vol. II), 149-166, San Mateo: Morgan Kaufmann.

Quinlan, J.R. (1989) Unknown Attribute values in induction, Proceedings of the Sixth International Workshop on Machine Learning, 164-168, Ithaca, NY: Morgan Kaufmann.

Rendell, L., Benedict, P., Cho, H., & Seshu, R. (1987) Improving the Design of Rule-learning Systems, Department of Computer Science, University of Illinois, UIUCDCS-R-87-1395.

Schlimmer, J.C., & Fisher, D. (1986) A case study of incremental concept induction. Proceedings of the Fifth National Conference on Artificial Intelligence, 496-501. Philadelphia, PA: Morgan Kaufmann.

Shafer, G., (1976) A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ.

Utgoff, P., (1988) ID5: An Incremental ID3, Proceeding of the International Conference on Machine Learning, 107-120, Ann Arbor, MI.

Utgoff, P.E. (1989) Incremental Learning of decision trees, Machine Learning, 4, 161-186.

Weiss, S., & Kapouless, I. (1989) An empirical comparison of pattern recognition, neural nets, and machine learning classification methods, Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 781-787, Detroit: Morgan Kaufmann.

Additional References

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984) Classification and Regression Trees, Belmont, CA: Wadsworth International Group.

Buntine, W. (1989) Learning Classification Rules Using Bayes, Proceedings of the Sixth International Workshop on Machine Learning, 94-98, Ithaca, NY: Morgan Kaufmann.

Chan, K.C.C., & Wong, A.K.C. (1990) Performance Analysis of a Probabilistic Learning System, Proceedings of the Seventh International Conference on Machine Learning, 16-23.

Dietterich, T.G. (1989) Limitations on Inductive Learning, Proceedings of the Sixth International Workshop on Machine Learning, 124-128, Ithaca, NY: Morgan Kaufmann.

Fisher, D.H., and McKusick, K.B. (1986) An Empirical Comparison of ID3 and Back-Propagation. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 788-793, Detroit: Morgan Kaufmann.

Fisher, D., McKusick, K., Mooney, R., Shavlik, J.W., and Towell, G., (1989) Processing Issues in Comparisons of Symbolic and Connectionist Learning Systems, Proceedings of the Sixth International Machine Learning Workshop, Ithaca, NY: Morgan Kaufmann.

Gams M., & Karalic, A. (1989) New Empirical Learning Mechanisms Perform Significantly Better in Real Life Domains, Proceedings of the Sixth International Workshop on Machine Learning, 99-103, Ithaca, NY: Morgan Kaufmann.

Haussler, D. (1988) Quantifying inductive bias: AI Learning algorithms and Valiant's learning framework, Artificial Intelligence, 36, 177-222.

Hirsh, H. (1990) Learning from Data with Bounded Inconsistency, Proceedings of the Seventh International Conference on Machine Learning, 32-39.

Kibler, D., & Langley, P. (1988) Machine Learning as an Experimental Science, Proceeding of the Third European Working Session on Learning, 81-92, San Mateo: Morgan Kaufmann.

Michalski, R.S. (1983) A Theory and Methodology of Inductive Learning, Artificial Intelligence, 20, 111-116.

Michalski, R.S., & Chilausky, C., (1980) Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis, International Journal of Policy Analysis and Information Systems, 4, 125-161.

Mooney, R.J., Shavlik, J.W., Towell, G.G., & Gove, A. (1989) An experimental comparison of symbolic and connectionist learning algorithms. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 775-780, Detroit: Morgan Kaufmann.

Pitt, L., & Valiant, L.G. (1988) Computational limitations on learning from examples, Journal of the ACM, 35, 965-84.

Quinlan , J.R. (1987) Simplifying decision trees, International Journal of Man-Machine Studies, 27, 221-234.

Shafer, G., & Logan, R. (1987) Implementing Dempster's Rule for Hierarchical Evidence, Artificial Intelligence, 33, 271-298.

Shenoy, P.P., & Shafer, G. (1986) Propagating Belief Functions with Local Computations, IEEE Expert, 1, 43-52.

Spangler, S., Fayyad, U.M., & Uthurusamy, R. (1990) Induction of Decision Trees from Inconclusive Data, Proceedings of the Sixth International Workshop on Machine Learning, 146-150, Ithaca, NY: Morgan Kaufmann.

Valiant, L.G., (1984) A theory of the learnable, Communications of the ACM, 27, 1134-1142.

Vroobraak, F. (1991) On the justification of Dempster's rule of combination, Artificial Intelligence, 48, 171-197.

Table I. Brackets for IRIS Dataset

```
Attributes
4
  seplen integer
   24    43-48 49-49 50-50 51-51 52-52 53-53 54-54 55-55 56-56
         57-57 58-58 59-59 60-60 61-61 62-62 63-63 64-64 65-65
         66-66 67-67 68-68 69-69 70-70 71-79
  sepwid integer
   19    20-21 22-22 23-23 24-24 25-25 26-26 27-27 28-28 29-29
         30-30 31-31 32-32 33-33 34-34 35-35 36-36 37-37 38-38
         39-44
  petlen integer
    9    10-29 30-44 45-45 46-47 48-48 49-49 50-50 51-51 52-69
  petwid integer
    8    01-9 10-13 14-14 15-15 16-16 17-17 18-18 19-25
Attributes_end
```

Table II. PAA for IRIS example

Rule Set (Raw Matrix)

| seplen | set | ver | vir |
|---|---|---|---|
| 43-48 | 16 | 0 | 0 |
| 49-49 | 4 | 1 | 1 |
| 50-50 | 8 | 2 | 0 |
| 51-51 | 8 | 1 | 0 |
| 52-52 | 3 | 1 | 0 |
| 53-53 | 1 | 0 | 0 |
| 54-54 | 5 | 1 | 0 |
| 55-55 | 2 | 5 | 0 |
| 56-56 | 0 | 5 | 1 |
| 57-57 | 2 | 5 | 1 |
| 58-58 | 1 | 3 | 3 |
| 59-59 | 0 | 2 | 1 |
| 60-60 | 0 | 4 | 2 |
| 61-61 | 0 | 4 | 2 |
| 62-62 | 0 | 2 | 2 |
| 63-63 | 0 | 3 | 6 |
| 64-64 | 0 | 2 | 5 |
| 65-65 | 0 | 1 | 4 |
| 66-66 | 0 | 2 | 0 |
| 67-67 | 0 | 3 | 5 |
| 68-68 | 0 | 1 | 2 |
| 69-69 | 0 | 1 | 3 |
| 70-70 | 0 | 1 | 0 |
| 71-79 | 0 | 0 | 12 |

| sepwid | set | ver | vir |
|---|---|---|---|
| 20-21 | 0 | 1 | 0 |
| 22-22 | 0 | 2 | 1 |
| 23-23 | 1 | 3 | 0 |
| 24-24 | 0 | 3 | 0 |
| 25-25 | 0 | 4 | 4 |
| 26-26 | 0 | 3 | 2 |
| 27-27 | 0 | 5 | 4 |
| 28-28 | 0 | 6 | 8 |
| 29-29 | 1 | 7 | 2 |
| 30-30 | 6 | 8 | 12 |
| 31-31 | 4 | 3 | 4 |
| 32-32 | 5 | 3 | 5 |
| 33-33 | 2 | 1 | 3 |
| 34-34 | 9 | 1 | 2 |
| 35-35 | 6 | 0 | 0 |
| 36-36 | 3 | 0 | 1 |
| 37-37 | 3 | 0 | 0 |
| 38-38 | 4 | 0 | 2 |
| 39-44 | 6 | 0 | 0 |

| petlen | set | ver | vir |
|---|---|---|---|
| 10-29 | 50 | 0 | 0 |

| | | | |
|---|---|---|---|
| 30-44 | 0 | 29 | 0 |
| 45-45 | 0 | 7 | 1 |
| 46-47 | 0 | 8 | 0 |
| 48-48 | 0 | 2 | 2 |
| 49-49 | 0 | 2 | 3 |
| 50-50 | 0 | 1 | 3 |
| 51-51 | 0 | 1 | 7 |
| 52-69 | 0 | 0 | 34 |

| petwid | set | ver | vir |
|---|---|---|---|
| 01-9 | 50 | 0 | 0 |
| 10-13 | 0 | 28 | 0 |
| 14-14 | 0 | 7 | 1 |
| 15-15 | 0 | 10 | 2 |
| 16-16 | 0 | 3 | 1 |
| 17-17 | 0 | 1 | 1 |
| 18-18 | 0 | 1 | 11 |
| 19-25 | 0 | 0 | 34 |

Rule Set (Converted into m)

| seplen | set | ver | vir | Theta |
|---|---|---|---|---|
| 43-48 | 0.320000 | 0.000000 | 0.000000 | 0.680000 |
| 49-49 | 0.040000 | 0.000000 | 0.000000 | 0.960000 |
| 50-50 | 0.120000 | 0.000000 | 0.000000 | 0.880000 |
| 51-51 | 0.140000 | 0.000000 | 0.000000 | 0.860000 |
| 52-52 | 0.040000 | 0.000000 | 0.000000 | 0.960000 |
| 53-53 | 0.020000 | 0.000000 | 0.000000 | 0.980000 |
| 54-54 | 0.080000 | 0.000000 | 0.000000 | 0.920000 |
| 55-55 | 0.000000 | 0.060000 | 0.000000 | 0.940000 |
| 56-56 | 0.000000 | 0.080000 | 0.000000 | 0.920000 |
| 57-57 | 0.000000 | 0.040000 | 0.000000 | 0.960000 |
| 58-58 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 59-59 | 0.000000 | 0.020000 | 0.000000 | 0.980000 |
| 60-60 | 0.000000 | 0.040000 | 0.000000 | 0.960000 |
| 61-61 | 0.000000 | 0.040000 | 0.000000 | 0.960000 |
| 62-62 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 63-63 | 0.000000 | 0.000000 | 0.060000 | 0.940000 |
| 64-64 | 0.000000 | 0.000000 | 0.060000 | 0.940000 |
| 65-65 | 0.000000 | 0.000000 | 0.060000 | 0.940000 |
| 66-66 | 0.000000 | 0.040000 | 0.000000 | 0.960000 |
| 67-67 | 0.000000 | 0.000000 | 0.040000 | 0.960000 |
| 68-68 | 0.000000 | 0.000000 | 0.020000 | 0.980000 |
| 69-69 | 0.000000 | 0.000000 | 0.040000 | 0.960000 |
| 70-70 | 0.000000 | 0.020000 | 0.000000 | 0.980000 |
| 71-79 | 0.000000 | 0.000000 | 0.240000 | 0.760000 |

| sepwid | set | ver | vir | Theta |
|---|---|---|---|---|
| 20-21 | 0.000000 | 0.020000 | 0.000000 | 0.980000 |

| | | | | |
|---|---|---|---|---|
| 22-22 | 0.000000 | 0.020000 | 0.000000 | 0.980000 |
| 23-23 | 0.000000 | 0.040000 | 0.000000 | 0.960000 |
| 24-24 | 0.000000 | 0.060000 | 0.000000 | 0.940000 |
| 25-25 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 26-26 | 0.000000 | 0.020000 | 0.000000 | 0.980000 |
| 27-27 | 0.000000 | 0.020000 | 0.000000 | 0.980000 |
| 28-28 | 0.000000 | 0.000000 | 0.040000 | 0.960000 |
| 29-29 | 0.000000 | 0.080000 | 0.000000 | 0.920000 |
| 30-30 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 31-31 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 32-32 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 33-33 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 34-34 | 0.120000 | 0.000000 | 0.000000 | 0.880000 |
| 35-35 | 0.120000 | 0.000000 | 0.000000 | 0.880000 |
| 36-36 | 0.040000 | 0.000000 | 0.000000 | 0.960000 |
| 37-37 | 0.060000 | 0.000000 | 0.000000 | 0.940000 |
| 38-38 | 0.040000 | 0.000000 | 0.000000 | 0.960000 |
| 39-44 | 0.120000 | 0.000000 | 0.000000 | 0.880000 |

| petlen | set | ver | vir | Theta |
|---|---|---|---|---|
| 10-29 | 1.000000 | 0.000000 | 0.000000 | 0.000000 |
| 30-44 | 0.000000 | 0.580000 | 0.000000 | 0.420000 |
| 45-45 | 0.000000 | 0.120000 | 0.000000 | 0.880000 |
| 46-47 | 0.000000 | 0.160000 | 0.000000 | 0.840000 |
| 48-48 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 49-49 | 0.000000 | 0.000000 | 0.020000 | 0.980000 |
| 50-50 | 0.000000 | 0.000000 | 0.040000 | 0.960000 |
| 51-51 | 0.000000 | 0.000000 | 0.120000 | 0.880000 |
| 52-69 | 0.000000 | 0.000000 | 0.680000 | 0.320000 |

| petwid | set | ver | vir | Theta |
|---|---|---|---|---|
| 01-9 | 1.000000 | 0.000000 | 0.000000 | 0.000000 |
| 10-13 | 0.000000 | 0.560000 | 0.000000 | 0.440000 |
| 14-14 | 0.000000 | 0.120000 | 0.000000 | 0.880000 |
| 15-15 | 0.000000 | 0.160000 | 0.000000 | 0.840000 |
| 16-16 | 0.000000 | 0.040000 | 0.000000 | 0.960000 |
| 17-17 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 18-18 | 0.000000 | 0.000000 | 0.200000 | 0.800000 |
| 19-25 | 0.000000 | 0.000000 | 0.680000 | 0.320000 |

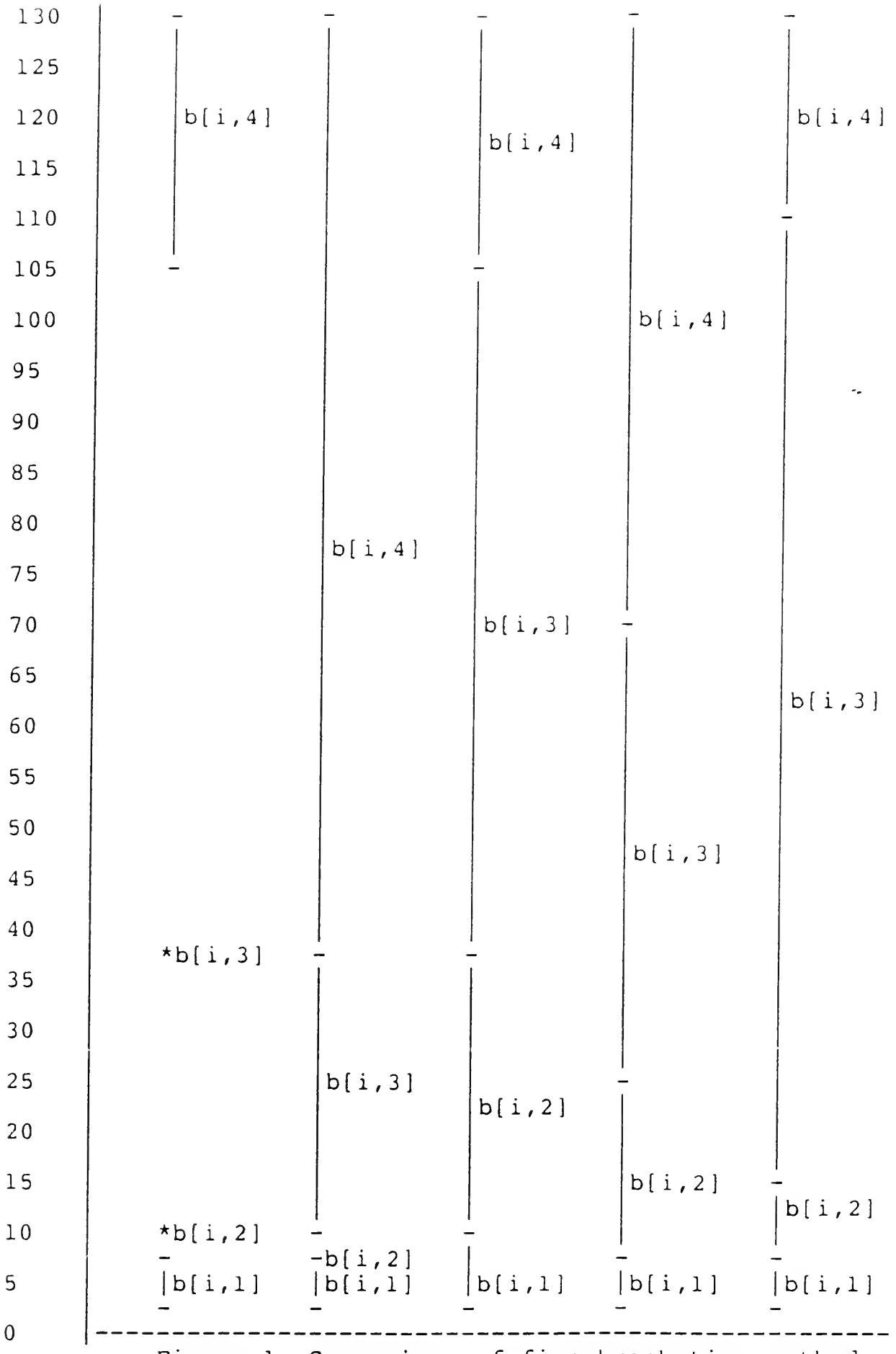


Figure 1. Comparison of five bracketing methods

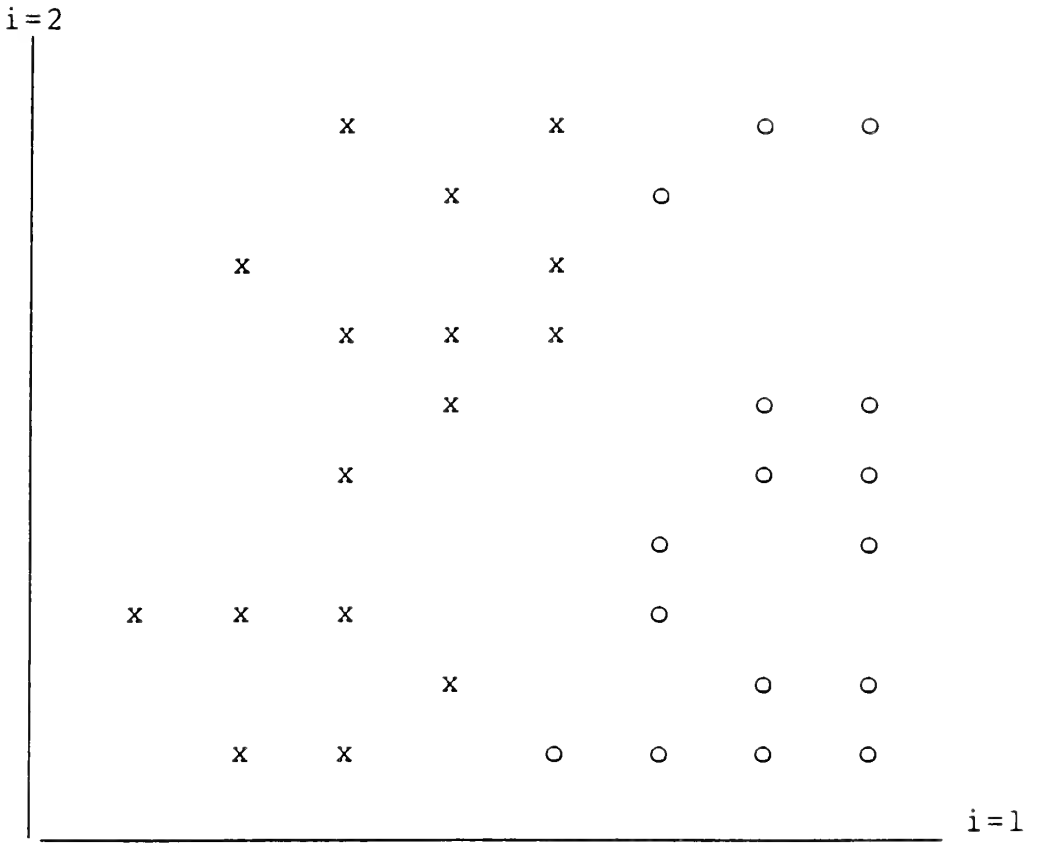


Figure 2. Example data distribution

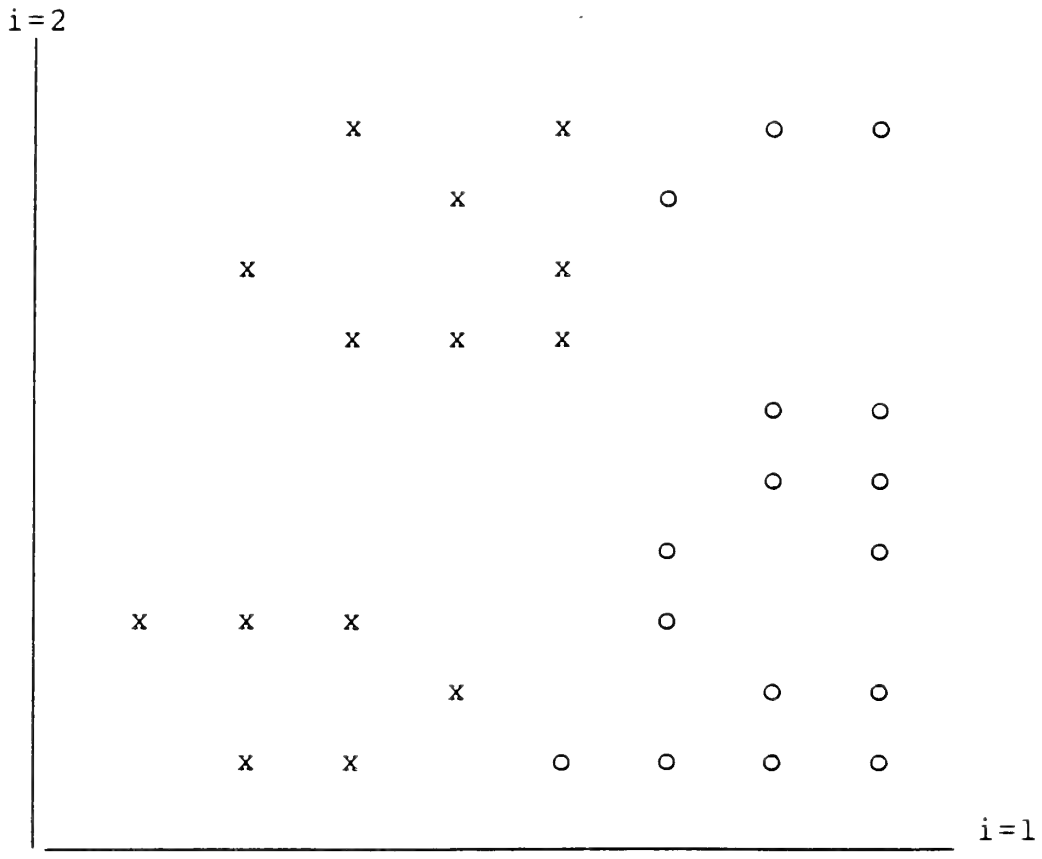


Figure 3. Regions defined by linear discriminant function

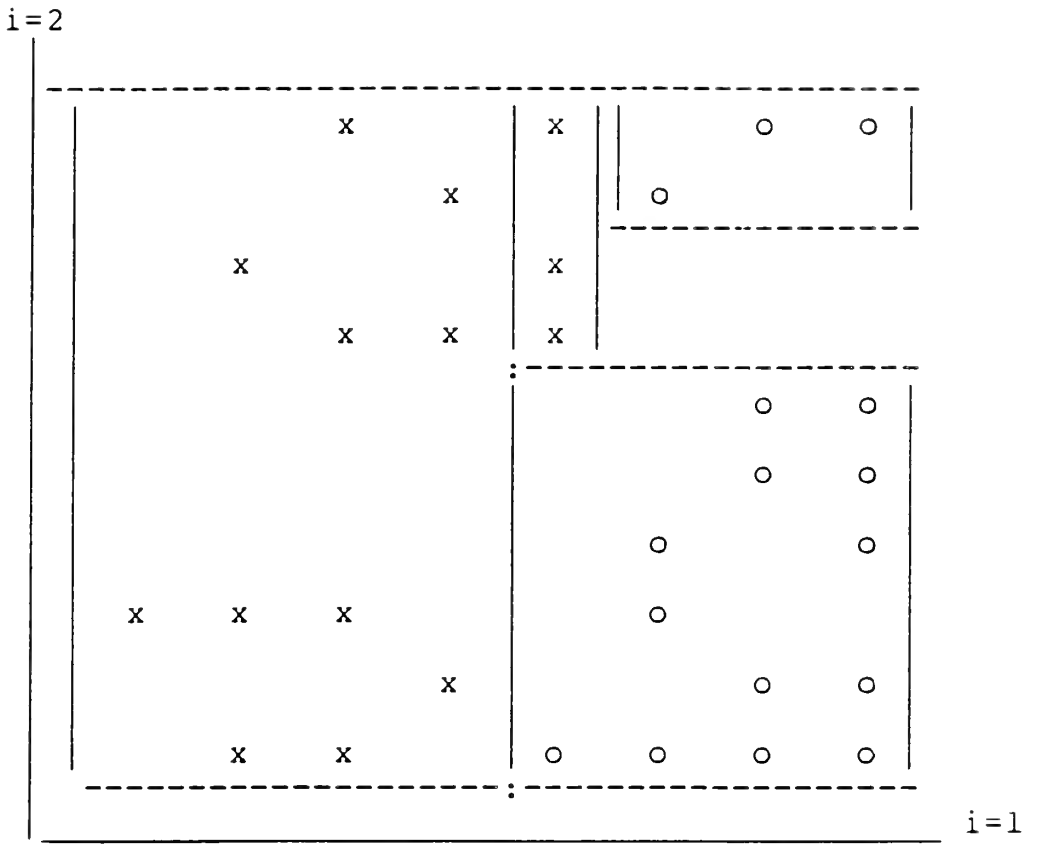


Figure 4. Rectangles formed by TDDT

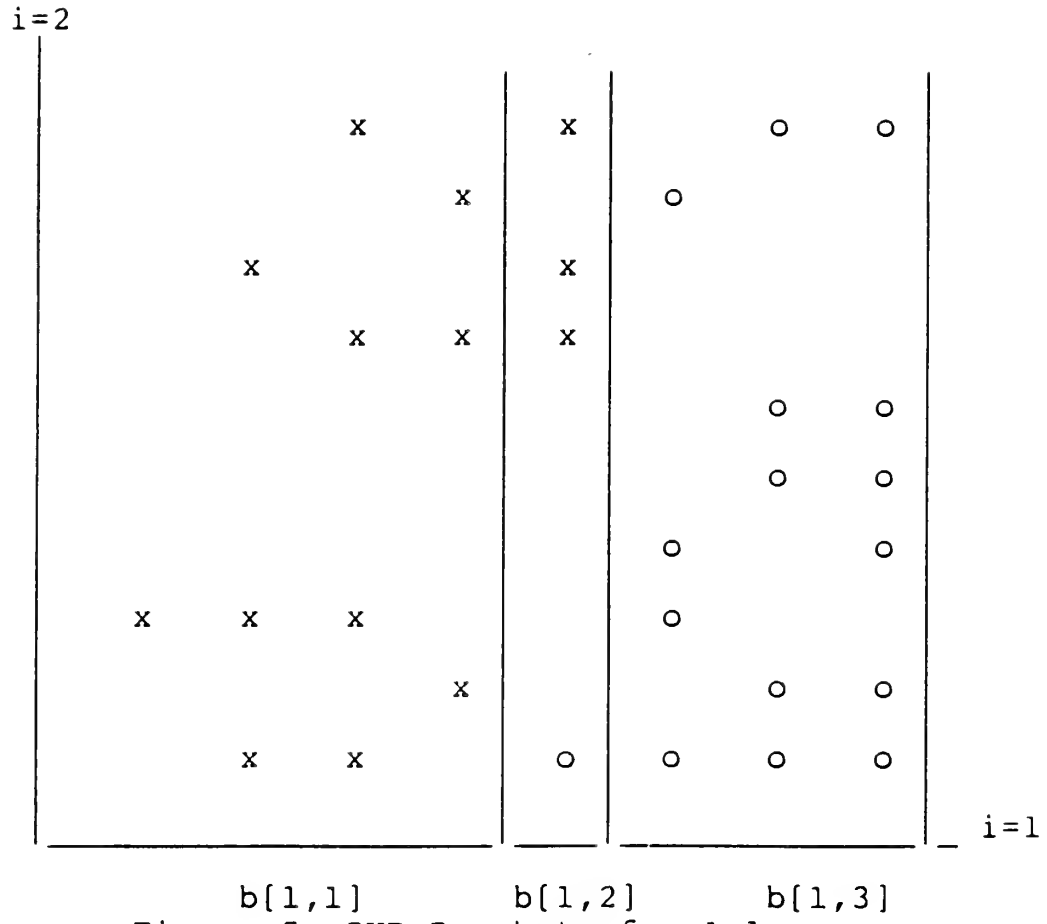

Figure 5. IUR Brackets for A.1

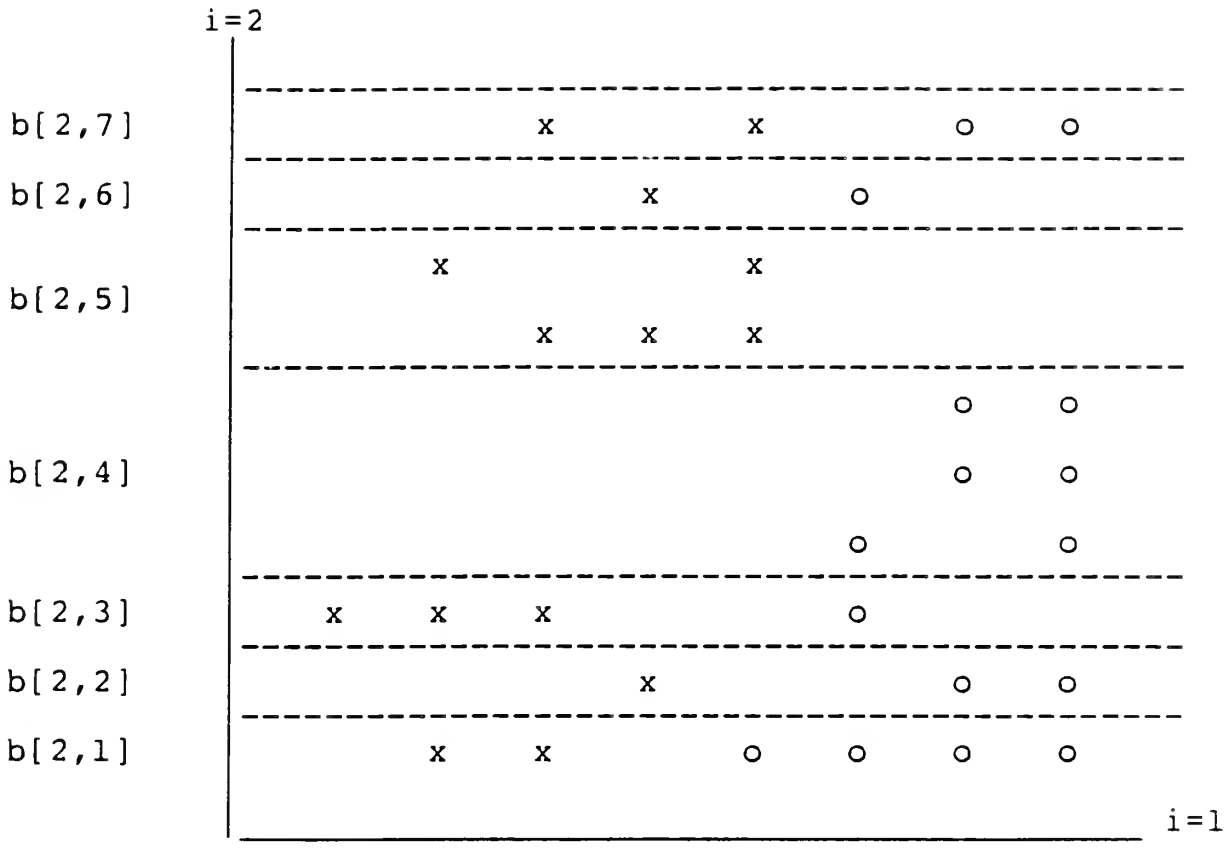


Figure 6. IUR Brackets for A.2

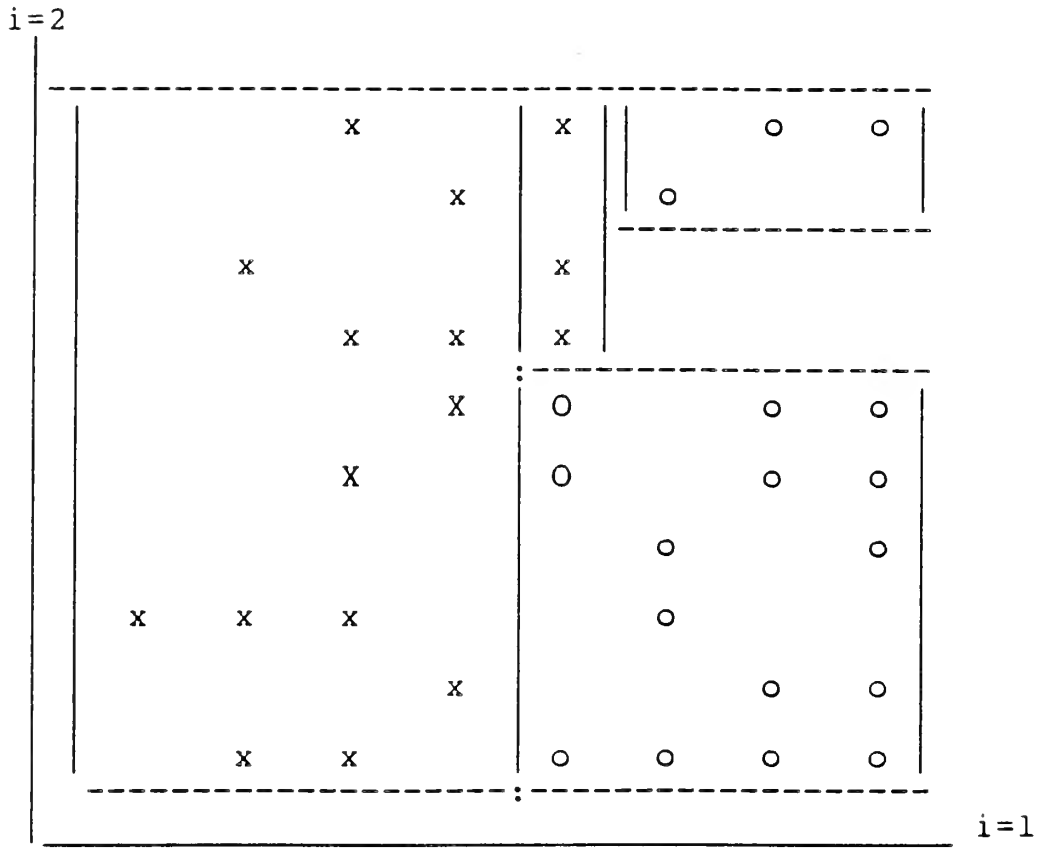


Figure 7. TDDT has same rectangles with different distribution

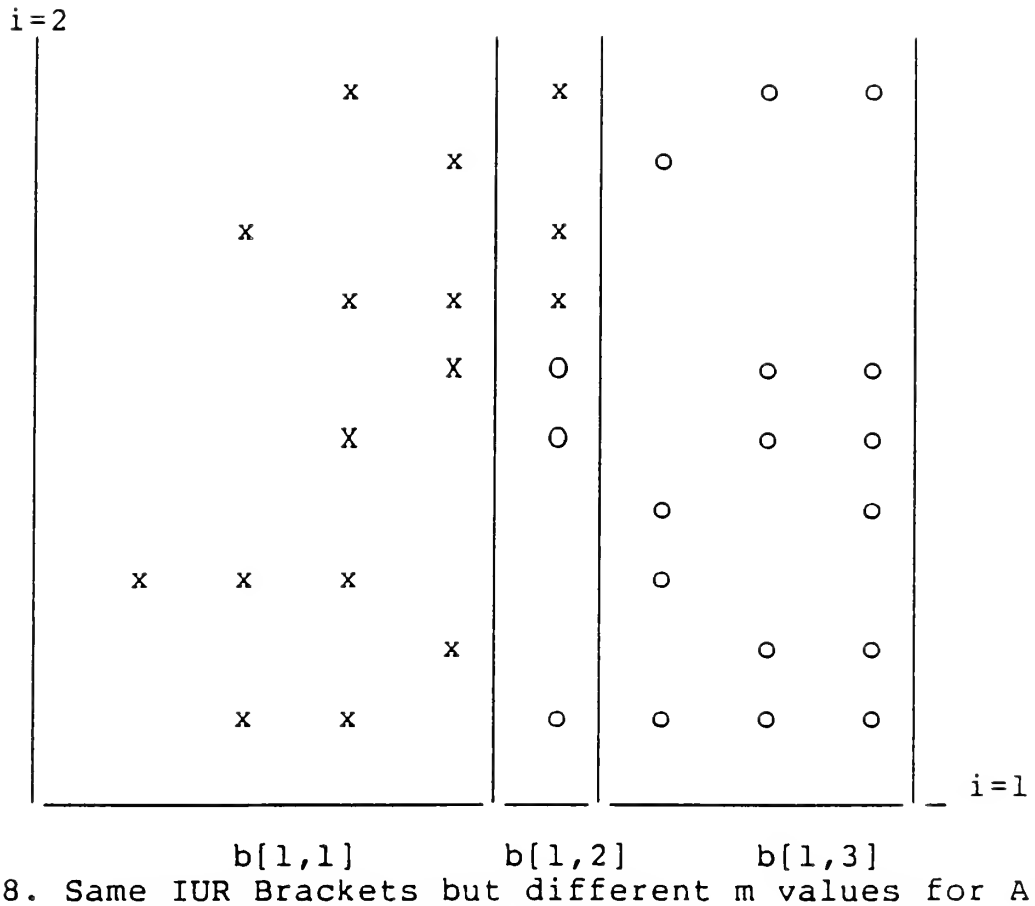


Figure 8. Same IUR Brackets but different m values for A.1 for different distribution

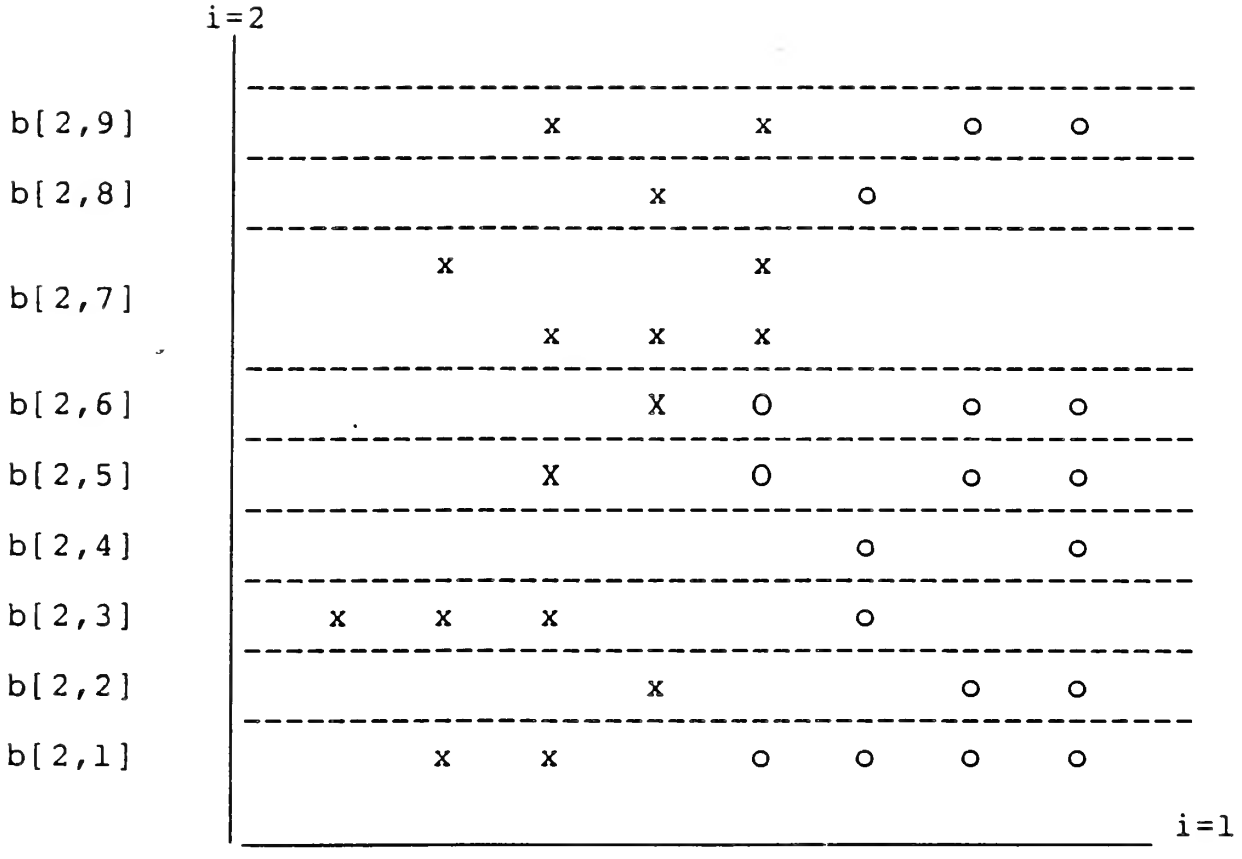


Figure 9. Different IUR Brackets and m values for A.2 for different distribution